~~MONITORING SERVER PERFORMANCE~~BACKGROUND OF THE INVENTION

~~Background of the Invention~~

~~The invention is directed to monitoring the performance of a server and a network.~~

Related Application

This application is a continuation of U.S. Application Serial No. 09/353,304 entitled "System and Method for Monitoring Server Performance Using a Server" filed July 13, 1999, which is a continuation-in-part of U.S. Application Serial No. 08/682,832 entitled "System and Method for Monitoring Server Performance at a Client Computer" (as amended), filed July 12, 1996, which is incorporated herein by reference in its entirety.

The Field of the Invention

The present invention relates to electronic communications. Specifically, the present invention relates to a system and method for monitoring server performance using a server.

The Prior State of the Art

~~The server runs server software that includes one or more software modules, i.e., services. The server~~Computer networks are playing an increasingly important role in the information age. One important component of a computer network is a server system. A "server system" is a hardware and/or software component that provides services to another component of the computer network typically termed a "client system." The client system may be, for example, a personal computer or workstation associated with a user. A typical server system may provide numerous services to a client system. The services might be available, for example, in the form of software modules. The server; software is supported by operating system software that provides the server software with access to resources such as the server's computer hardware resources and the server's user interface resources. It is desirable to monitor the performance of

server systems on a network to ensure the network is operating properly. Typically, the operating status of an individual service of a server system may be determined by sending a computer-generated query to the ~~operating~~server system. In response to the query, the ~~operating~~server system sends a reply that includes the operating status of the service. The reply indicates, among other things, whether the service is running. The ~~query may be sent by a computer that gains access to the server across a computer data connection such as a network connection. To~~queries and replies typically include at least one query and one reply per service. Thus, to determine the status of multiple services at ~~one~~a single server, the querying computer sends multiple queries and receives multiple replies. The ~~queries and replies typically include at least one query and one reply per service~~issuance of each query takes time. Thus, it is desirable to reduce the number of queries generated to monitor the performance of the services provided by a server system.

~~The server software may be included in a messaging system that allows a user to direct a message from a source location, e.g., a messaging server, on the messaging system to a destination location, e.g., another messaging server, on the messaging system. Typically, each user of the messaging system is associated with a set of resources in the messaging system, i.e., has a messaging account. Each time the user initiates a messaging session for, e.g., reading and otherwise manipulating messages, the user is typically required to execute a "login" procedure. The login procedure provides the user with access to the user's messaging account after verifying that the user has permission to use the account. In the verification, the user is typically required to provide at least a proper password.~~

~~Each location in the messaging system typically makes use of a time indicator that is provided at the location, typically by the operating system. The time indicator is used by the location for purposes such as including a timestamp in an outgoing message, to indicate when the~~

outgoing message was sent. Each location's time indicator operates independently of the time indicators of other locations and is adjusted independently as well.

## Summary of the Invention

A technique is provided wherein monitoring software is divided between a monitored server and a monitoring user computer, with the server portion of the software configured to reply, to a query from the monitoring user computer, with a compilation of information about the server and software services running on the server. Queries are sent periodically and are separated by a normal time interval or a shorter, critical time interval. The queries proceed according to an escalation procedure in which the monitoring user computer automatically switches to using the critical interval after a deficiency is discovered. In addition, the server portion of the software periodically determines whether a user of messaging software on the server is able to gain access to the user's messages. The outcome of this determination is preferably included in the reply to each query. A time indicator that is located at the server and that is used by the messaging software is monitored to determine the accuracy of the time indicator with respect to a master time indicator that is located at the monitoring user computer.

The invention has several advantages. Information about the server is sent in a single compilation, allowing the monitoring user computer to avoid having to send repeated queries to the server to gather necessary information from the server. Servers that are not connected to the network, i.e., are without power or are not running network connectivity software properly, can be detected. Subsequent to a discovery of a deficiency on the server, the frequency of providing information about the server is increased without intervention by an administrator. The user computer portion of the software can monitor multiple servers. The selection of the server's services to be monitored by the monitoring software is determined by the server portion of the monitoring. software; thus, changing the selection of the services to be monitored on a server requires no change to the user computer portion of the monitoring software. The clocks of

servers in a messaging system are synchronized so that message timestamps are meaningful within the messaging system. The use of network resources in the messaging system is optimized during monitoring by re-establishing a working connection in the system only after a detection of a deficiency in the connection.

In one aspect, the invention feature a method of monitoring a server computer, the method including compiling at the server computer a collection of information about the status of the server computer, the collection including a plurality of status indicators, sending a query to the server computer from a user computer, and sending a reply from the server computer to the user computer, the reply including the collection.

Implementations of this aspect of the invention may include one or more of the following features.

The collection may include the status of a software service at the server. A messaging system, a facsimile communications system, or an Internet communications system may include the software service. The software service may include a directory service, a message store service, or a message transfer agent service. The collection may include information about a deficiency at the server. The reply may include information directed to be included in the reply by a configuration setting at the server. The user computer may send a query to each of a plurality of servers and may receive a reply from each of the plurality of servers.

In another aspect, the invention features a method of monitoring a server computer, the method including sending a plurality of queries for information about the server computer, the queries being sent at an adjustable frequency, each query being sent to the server computer from a user computer, compiling the information in a reply to one of the queries, sending the reply from the server computer to the user computer and, based on the reply, adjusting the frequency

of sending the queries. Implementations of this aspect of the invention may include one or more of the following features. The reply may include information about a deficiency at the server computer and the frequency may be increased as a result of the deficiency. The reply may include the status of a software service at the server computer or information directed to be included in the reply by a configuration setting at the server computer.

The method may further include using a plurality of lists of server computers, wherein each list corresponds to a different frequency of sending queries. The method may further include determining from the reply whether a deficiency exists at the server computer and depending on the outcome of that determination, moving a list entry for the server computer, the list entry being moved from one of the lists of server computers to another of the lists of server computers. Moving the list entry may further depend on a configuration setting.

In yet another aspect, the invention features a method of monitoring a messaging system, the method including automatically attempting to establish a working connection to a mailbox in the messaging system, determining whether the working connection *is* established, and reporting the results of the attempt.

Implementations of this aspect of the invention may include one or more of the following features. Depending on the outcome of the determination, the method may further include automatically attempting to determine whether a message is available at the mailbox and may report the results of the attempt to determine whether a message is available. Depending on the outcome of the determination about whether a message is available, the method may further include attempting to read the message and reporting the results of the attempt to read the message.

In still another aspect, the invention features a method of monitoring a messaging system,

the method including obtaining a time value from a time indicator associated with a server in the messaging system, comparing the time value with a master time value from a master time indicator, and based on the outcome of the comparison, updating the time indicator.

Implementations of this aspect of the invention may include the following feature. The time value may include a time zone indicator and the method may further include determining a difference between the time value and the master time value, the determination using the time zone indicator, and comparing the difference to a maximum difference drawn from a configuration setting. The technique may be implemented in hardware or software, or a combination of both. Preferably, the technique is implemented in computer programs executing on programmable computers that each include a processor, a storage medium readable by the processor (including volatile and non-volatile memory and/or storage elements), at least one input device, and at least one output device. Program code is applied to data entered using the input device to perform the method described above and to generate output information. The output information is applied to one or more output devices. Each program is preferably implemented in a high level procedural or object oriented programming language to communicate with a computer system. However, the programs can be implemented in assembly or machine language, if desired. In any case, the language may be a compiled or interpreted language.

Each such computer program is preferably stored on a storage medium or device (e.g., ROM or magnetic diskette) that is readable by a general or special purpose programmable computer for configuring and operating the computer when the storage medium or device is read by the computer to perform the procedures described in this document. The system may also be considered to be implemented as a computer-readable storage medium, configured with a

computer program, where the storage medium so configured causes a computer to operate in a specific and predefined manner. Other features and advantages will become apparent from the following description, including the drawings, and from the claims.

## Brief Description of the Drawings

FIG. 1 is a block diagram of a messaging system.

## SUMMARY OF THE INVENTION

The present invention relates to a method and system for monitoring the performance of a monitored server system using a polling server system. However, the number of queries needed to be issued for each monitored server system is significantly reduced, thereby significantly reducing polling time. Instead of a query being generated and transmitted for each service offered by a monitored server system, a single query is generated for the entire monitored server system. Specifically, the single query is generated by a polling server system. The single query represents a request for a compilation of information about the performance of numerous services offered by the monitored server system. After generating the single query, the polling server system then transmits this single query to the monitored server system. Upon receiving the single query, the monitored server system generates the compilation of information regarding the services it offers, and then transmits the compilation of information to the polling server system. Thus, with this single query, the polling server system obtains an entire compilation of information regarding the performance of the monitored server system. The polling server can likewise query other monitored server systems to obtain further compilations.

The compilation of information may include performance deficiencies detected in the monitored server system. When a deficiency is detected in the most recent polling of the monitored server system, the monitored server system is listed in a critical server system list.

At this point, the polling server system may poll the critical server system more frequently since the monitoring of the server system performance is typically more important once an operating deficiency is detected. Once the compilation of information is obtained by the polling server system, any or all of the compilation may be then provided to a client system so that, for example, a system administrator can review the information to diagnose and remedy any

deficiency in the monitored server system. The compilation may be provided to the client system either automatically, or upon the happening of a predetermined event such as receiving a request for the compilation from the client system. Additional objects and advantages of the invention will be set forth in the description which follows, and in part will be obvious from the description, or may be learned by the practice of the invention. The objects and advantages of the invention may be realized and obtained by means of the instruments and combinations particularly pointed out in the appended claims. These and other objects and features of the present invention will become more fully apparent from the following description and appended claims, or may be learned by the practice of the invention as set forth hereinafter.

## BRIEF DESCRIPTION OF THE DRAWINGS

In order that the manner in which the above-recited and other advantages and objects of the invention are obtained, a more particular description of the invention briefly described above will be rendered by reference to specific embodiments thereof which are illustrated in the appended drawings. Understanding that these drawings depict only typical embodiments of the invention and are not therefore to be considered limiting of its scope, the invention will be described and explained with additional specificity and detail through the use of the accompanying drawings in which:

Figure 1 illustrates an exemplary system that provides a suitable operating environment for the present invention;

Figure 2 is a schematic diagram illustrating a suitable network environment in which the invention can be implemented;

~~FIG. 2~~Figure 3 is a ~~flow chart~~flowchart of a ~~procedure~~method for polling ~~servers for information.~~ monitored server systems for information;

~~FIG. 3 is a block diagram of a server and a user computer in a messaging system.~~

~~FIG. 4 is a flow chart of a procedure for testing the mail-reading and login execution~~Figure 4 is a block diagram of a client-server system in the network environment of Figure 2; Figure 5 is a flowchart of a method for the login and mail-reading capabilities of ~~a messaging~~the monitored server~~.~~ system;

~~FIG. 5~~Figure 6 is a ~~flow chart~~flowchart of a ~~procedure~~method for testing the buffer-service-updating capability of ~~a messaging~~the monitored server~~.~~ system;

~~FIG. 6~~Figure 7 is a block diagram of ~~a construction~~the structure of a compilation reply~~.~~;

~~FIG. 7~~Figure 8 is a ~~flow chart~~flowchart of a procedure for including the status of a software service in ~~a~~the compilation reply~~.~~ of Figure 7; and

~~FIG. 8~~Figure 9 is a ~~flow chart~~flowchart of a procedure for synchronizing a monitored server system time indicator ~~and~~with a master time indicator.

## Description of the Preferred Embodiments

FIG. 1 illustrates a messaging system that includes messaging servers and user computers that are linked to form a networked computer system such as a client server system. To use the messaging system, a user preferably employs user messaging software running on a user computer. In combination with server messaging software running on a server, the user software allows the user to read messages stored at the user's mailbox at the server, create outgoing messages, and otherwise manipulate messages. For reasons such as m1n1mizing the user software's use of the link with the server software, the user software preferably establishes a working connection with the server software only under certain circumstances. For example, the working connection is established when the user initiates a messaging session to manipulate messages. To initiate a messaging session, the user activates the user software and causes the user software to execute a login procedure. A password for the user is passed to the server software, which allows such a working connection to be established only if the server recognizes the password. After the login procedure is completed, the user is able to proceed with the messaging session. When the messaging session is over, the working connection is preferably broken by the user, when the user de-activates the user software.

## DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

The present invention relates to systems and methods for monitoring server system performance using a server system. Specifically, a polling server system generates a single request to obtain a compilation of information from a monitored server system. Since fewer queries are needed to monitor each server system, polling time is reduced and network efficiency improved. Embodiments within the scope of the present invention include computer-readable media having computer-executable instructions or data structures stored thereon. Such computer-readable media can be any available media which can be accessed by a general purpose or special purpose computer. By way of example, and not limitation, such computer-readable media can comprise RAM, ROM, EEPROM, CD-ROM or other optical disk storage, magnetic disk storage or other magnetic storage devices, or any other medium which can be used to store the desired computer-executable instructions or data structures and which can be accessed by a general purpose or special purpose computer. When information is transferred or provided over a network or another communications connection to a computer, the computer properly views the connection as a computer-readable medium. Thus, such a connection is also properly termed a computer-readable medium. Combinations of the above should also be included within the scope of computer-readable media. Computer-executable instructions comprise, for example, instructions and data which cause a general purpose computer, special purpose computer, or special purpose processing device to perform a certain function or group of functions. Figure 1 and the following discussion are intended to provide a brief, general description of a suitable network environment in which the invention may be implemented. Although not required, the invention will be described in the general context of computer-executable instructions, such as program modules, being executed by computers in network

environments. Generally, program modules include routines, programs, objects, components, data structures, etc. that perform particular tasks or implement particular abstract data types. Computer-executable instructions, associated data structures, and program modules represent examples of the program code means for executing steps and acts of the methods disclosed herein. Those skilled in the art will appreciate that the invention may be practiced in network computing environments with many types of computer system configurations, including personal computers, hand-held devices, multi-processor systems, microprocessor-based or programmable consumer electronics, network PCs, minicomputers, mainframe computers, and the like. The invention may also be practiced in distributed computing environments where tasks are performed by local and remote processing devices that are linked through a communications network. In a distributed computing environment, program modules may be located in both local and remote memory storage devices. With reference to Figure 1, an exemplary system for implementing the invention includes a general purpose computing device in the form of a computer, including a processing unit, a system memory , and a system bus that couples various system components including the system memory to the processing unit. The system bus may be any of several types of bus structures including a memory bus or memory controller, a peripheral bus, and a local bus using any of a variety of bus architectures. The system memory includes read only memory (ROM) and random access memory (RAM). A basic input/output system (BIOS), containing the basic routines that help transfer information between elements within the computer, such as during start-up, may be stored in ROM. The computer may also include a magnetic hard disk drive for reading from and writing to a magnetic hard disk, not shown, a magnetic disk drive for reading from or writing to a removable magnetic disk, and an optical disk drive for reading from or writing to removable optical disk such as a CD-ROM or other

optical media. The magnetic hard disk drive, magnetic disk drive, and optical disk drive are connected to the system bus by a hard disk drive interface, a magnetic disk drive-interface, and an optical drive interface, respectively. The drives and their associated computer-readable media provide nonvolatile storage of computer-executable instructions, data structures, program modules and other data for the computer. Although the exemplary environment described herein employs a magnetic hard disk, a removable magnetic disk and a removable optical disk, other types of computer readable media for storing data can be used, including magnetic cassettes, flash memory cards, digital video disks, Bernoulli cartridges, RAMs, ROMs, and the like. A number of program modules may be stored on the hard disk, magnetic disk, optical disk, ROM or RAM, including an operating system, one or more application programs, other program modules, and program data. A user may enter commands and information into the computer through keyboard, pointing device, or other input devices (not shown), such as a microphone, joy stick, game pad, satellite dish, scanner, or the like. These and other input devices are often connected to the processing unit through a serial port interface coupled to system bus. Alternatively, the input devices may be connected by other interfaces, such as a parallel port, a game port or a universal serial bus (USB). A monitor or another display device is also connected to system bus via an interface, such as video adapter. In addition to the monitor, personal computers typically include other peripheral output devices (not shown), such as speakers and printers. The computer may operate in a networked environment using logical connections to one or more remote computers, such as a remote computer. Remote computer may be another personal computer, a server, a router, a network PC, a peer device or other common network node, and typically includes many or all of the elements described above relative to the computer, although only a memory storage device has been illustrated in Figure 1. The logical

connections depicted in Figure 1 include a local area network (LAN) and a wide area network (WAN) that are presented here by way of example and not limitation. Such networking environments are commonplace in office-wide or enterprise-wide computer networks, intranets and the Internet. When used in a LAN networking environment, the computer is connected to the local network through a network interface or adapter. When used in a WAN networking environment, the computer typically includes a modem or other means for establishing communications over the wide area network, such as the Internet. The modem, which may be internal or external, is connected to the system bus via the serial port interface. In a networked environment, program modules depicted relative to the computer, or portions thereof, may be stored in the remote memory storage device. It will be appreciated that the network connections shown are exemplary and other means of establishing a communications link between the computers may be used. Figure 2 illustrates a client-server system that includes server systems that are networked together and with client systems. Client software resides on the client systems and server software, including various services, resides on the server systems. Each server system also may include a time indicator that is used by the server software for adding a timestamp to outgoing messages. In one embodiment, client-server system is a messaging system, server systems are messaging servers, client systems are user computers such as the computer of Figure 1, client software is client messaging software, and server software is server messaging software. Although this embodiment of a messaging system is described in further detailed below, it will be understood that the principles of the present invention may extend to many other types of network systems as well. In the case of a messaging system, the client messaging software and the server messaging software allow the user associated with each client system to create outgoing messages and otherwise manipulate messages. In addition, in order to

read messages stored at the user's mailbox at the corresponding server system, the user's client system links with the corresponding server system by, for example, initiating a messaging session. To initiate a messaging session, the user runs the client messaging software and causes the client messaging software to log into the server system using a login procedure. Typically, a user password is passed to the server messaging software, which allows a working connection to be established only if the server system recognizes the password. After the login procedure is completed, the user is able to proceed with the messaging session. The messaging session may be closed and the working connection may be broken by the user de-activating the client messaging software. Deficiencies in the client-server system are detected by a monitoring system that includes polling software that resides on one of the server systems that polls (namely, polling server system) and attendant software that resides on the server systems that are monitored (namely, monitored server systems). Each instance of attendant software compiles information about the server system on which the attendant software runs and, when queried by the polling software, provides the compilation to the polling software.

~~In addition, each server preferably provides a time indicator that is used by the server software for one or more purposes such as adding a timestamp to outgoing messages created by the user. Deficiencies in the messaging system are detected by a monitoring system that includes system attendant programs that report to polling software. Each server that is monitored by the monitoring system runs such an attendant. The polling software preferably runs on a user computer and provides an administrator with access to the monitoring system. As described below, each attendant compiles information about the server on which the attendant runs and, when queried by the polling software, provides the compilation to the polling software. The compilation preferably includes at least~~The compilation includes information about the services

of the server messaging software including, ~~e.g.~~ for example, a directory service, a message store service, and a message transfer agent ~~("MTA")~~ service. The directory service maintains records of locations of mailboxes in the messaging system and is used in the delivery of a message. The message store service stores and retrieves incoming messages for reading and other manipulation by users associated with the server. The ~~MTA~~message transfer agent service consults the directory service to deliver messages in the messaging system. The polling software ~~preferably~~ maintains two lists of server ~~sites, i.e., two lists of system attendants, to query: a normal sites list and a critical sites list. The sites in the normal sites~~ list are queried by the polling software in accordance with a "normal" polling interval, e.g., minutes. The polling software uses a shorter, "critical" polling interval, e.g., 2 minutes, for the sites in the critical sites list. Initially, all of the sites monitored by the monitoring systemsystems to query: a normal server system list and a critical server system list. The critical server system list includes all server systems that the polling software has discovered as having deficiencies. The normal server system list includes all the other monitored server systems. Initially, all of the server systems monitored by the polling software are included in the normal ~~sites list~~server system list since the server systems are assumed to have no deficiencies unless a deficiency is discovered. As described below, a ~~site~~server system entry may ~~be~~ transferred from the normal ~~sites~~server system list to the critical ~~sites~~server system list if a deficiency is discovered at the ~~site. In~~server system. The server systems in the normal server system list are queried by the polling software in accordance with a "normal" polling interval such as, for example, minutes. The server systems in the critical server system list are queries at more frequent intervals of, for example, minutes. Figure 3 is a diagram of a polling procedure ~~(FIG. 2)~~ that is triggered in ~~accordance with~~response to the commencement of one of the above-described polling intervals~~, the~~. The polling software first

determines the ~~sites~~server systems to query~~. The polling software so determines~~ by referring to the appropriate ~~sites~~server systems list as is now described. If the procedure was triggered in accordance with the critical polling interval (~~step 200~~Yes in decision block 300), the polling software refers to the critical ~~sites~~server system list (step ~~210~~310). Otherwise (No in decision block 300), the polling software refers to the normal ~~sites~~server system list (step ~~220~~320). ~~Preferably, the~~The polling software then ~~queries~~performs the ~~attendant~~following method for each ~~site listed~~server system in the ~~sites~~resulting list ~~(step 230). The~~. First, the polling software ~~so~~ queries ~~by~~the server system (step 330) by, for example, sending a network message to ~~each~~the attendant~~. Each~~ software associated with the monitored server system. The attendant software is expected to reply to the query with information in a compilation ~~that is described below. The information from each attendant~~. The reply includes status indicators that indicate whether deficiencies exist in the time indicator or ~~software~~services of the ~~site corresponding to the attendant. If an~~server system. If the attendant software for ~~a particular site~~the server system does not ~~respond~~reply with a compilation of information (~~step 240~~No in decision block 340), the polling software determines that the attendant software is not operating properly, which constitutes a deficiency. ~~The polling software then updates the sites lists, using the information about deficiencies, if any, and pre-selected configuration settings made by the administrator (step 250). The lists are preferably updated according to an escalation procedure. For example, for queries involving the normal sites list, the~~The non- responsive server system is then listed in the critical server system list (step 360). If the attendant software or the monitored server system does reply with a compilation of information (Yes in decision block 340), the polling software determines if there are any deficiencies in• the server system as indicated in the reply (decision block 350). If a deficiency is found (Yes in decision block 350), the server

system is listed in the critical server system list (step 360). On the other hand, if there are no deficiencies found (No in decision block 350), the server system is listed in the normal server system list. This process is repeated for all server systems in the list generated by steps 300, 310 and 320. For example, in the context of a messaging system, an administrator may configure ~~the polling software with list-transfer settings that are based on new deficiencies discovered (step 260). Such a list-transfer setting may direct~~ the polling software to transfer ~~the~~ list entry for ~~a site~~the server system to the critical ~~sites~~server system list if the polling software discovers that the site's directory service is in a non-functioning state (step ~~270~~360). Such a transfer is appropriate, because the monitored server system is unable to operate properly without a functioning directory service. Transferring the list entry for the ~~site~~deficient' server system to the critical ~~sites~~server system list ~~after such a discovery~~ allows the polling software to provide ~~the~~an administrator with information about the ~~site at more frequent, e.g., 2-minute, intervals.~~ deficient server system more frequently. The administrator is also able to pre-select configuration settings for the polling software for queries involving the critical ~~sites~~server system list. For example, according to a list-transfer setting, a list entry for a ~~site~~server system may be transferred back to the normal ~~sites~~server system list (step ~~280~~370). Such a transfer may occur if all deficiencies reflected in earlier replies from the ~~site~~server system's attendant software are absent from the most recent reply from ~~the~~that attendant software. In most cases, such a setting is appropriate, because, after all of a ~~site~~server system's deficiencies have cleared, the administrator typically desires information at less frequent~~, e.g., 15-minute,~~ intervals. A deficiency may clear in response to one or more actions taken by the administrator or by an automated mechanism or both. At the ~~site~~monitored server system, the attendant software preferably ~~gathers~~may gather some information autonomously and other information only after receiving a query from the

polling software. However, ~~preferably~~ all of the information provided in response to the query, regardless of the way in which some of the information is gathered, is provided to the polling software in one compilation and is not provided piecemeal. For example, the attendant software autonomously monitors a particular capability of the server messaging software.~~,~~ The capability involves allowing the reading of mail and making a working connection with a new instance of ~~user~~client messaging software, as described above. The attendant software so monitors by simulating the executions of a mail-reading procedure and a login procedure. The simulation is accomplished by exercising the server messaging software in the same way that the server messaging software is exercised in the executions of actual mail-reading and login procedures. ~~The~~ Figure 4 illustrates the operation of the server messaging software ~~is exercised as follows (FIG. 3)~~. The system attendant software has a messaging account, i.e., the attendant account, that is created when the attendant software is first located and started at the server system. The use of the attendant account differs from the use of the messaging account of an ordinary user of the server system in that the attendant software does not send a password to gain access to the attendant account. Instead, the attendant software gains access by submitting security codes retrieved from a network source that is responsible for network security. In other regards, the attendant account is ~~preferably~~preferable identical, aside from its contents and the like, to an ordinary user's account. At the time the attendant account is created, a message store buffer service is also located and started at the server system. Through the buffer service , the attendant software gains access to the server software's message store service. The buffer service is preferably identical to a user buffer service that, in the case of an actual user, runs on the user's ~~user computer~~client system along with the user messaging software. With respect to an actual user, the user buffer service minimizes the user software's use of the link between the user

software and server software by buffering information from the message store service. For example, when the message store service receives a new message directed to the user, running the userclient messaging software, the message store service sends a new-mail indication to the user buffer service. Thus, when the userclient messaging software is ready to determine whether the user has one or more new messages, the user software is able to make the determination by referring to the user buffer service, not to the message store service directly. The attendant software tests the use of the buffer service by executing, at periodic intervals "A" of preferably one minute each, an interval A function (FIG. 4).as shown in Figure 5. First, the attendant software determines whether it is necessary for the attendant software to execute a login procedure to establish a working connection with the server software (stepdecision block 400). If so, a login procedure is attempted (step 405). In the login procedure, the server software restarts services and establishes a working connection with the attendant software. If the login procedure fails (410), the attendant software reports a login failure (step 415) for the next compilation reply, as described below, and terminates the execution of the interval A function. On the other hand, if the login procedure is successful, the attendant software reports a login success (step 420) and the execution of the interval A function continues. A working connection having been established during this or a previous instance of executing an interval A function, the attendant software then requests that the buffer service to indicate whether any new messages have arrived for the attendant software (step 425). If the buffer service responds properly to the request (stepYes in decision block 430), the attendant software then determines whether the response of the buffer service's response indicates that the attendant software has a new message (step 435). If the response so indicates (Yes in decision block 435), the attendant software then attempts to read the new message (step 440). A successful read attempt or a lack of a new message to read

causes the attendant software to finish the interval A function with a successful report (step 445).

However, if the attendant software does not receive a proper response from the buffer service or if the read attempt is unsuccessful, the attendant software reports failure (step 450) and then executes a logoff procedure (step 455). In the logoff procedure, the attendant software terminates the working connection with the server software. The logoff procedure is executed to set up the attendant software to attempt to execute a login procedure at the start of the next execution of an interval A function. The logoff-login sequence is used to (1) attempt to address the cause of the failure, by causing, as described above, services to restart in the login procedure, and (2) test the ability of the server software to accept the execution of a login procedure.

In the case of an actual user and user computer, the buffer service rarely requests information from the message store service. Rather, the buffer service simply waits for the message store service to provide information. However, the buffer service includes a facility that is triggered when the buffer service has not received information from the message store service for a long time, e.g., one hour, while a working connection exists. The facility is triggered to confirm that the lack of received information is not due to a problem in communications between the buffer service and the message store service. The facility updates the buffer service's buffer of information by causing the buffer service to renew the buffer service's connection with the message store service and to request information from the message store service. The attendant software uses the facility at intervals "B", preferably every five minutes. The use of the facility at every five minutes allows the attendant software to detect a communications problem quickly, i.e., within much less than the facility's standard one-hour interval. In the interval B function (FIG. 5), of Figure 6, the attendant software is able to use the facility only if the attendant software has a working connection to the server software, i.e., if the attendant software is

"logged in" (~~step~~Yes in decision block 500). The attendant software invokes the facility (step 510) and then determines whether the invocation was successful (step 520). The invocation is expected to be successful unless the buffer service has developed a problem. If successfully invoked, the facility then requests an update, as described above, from the message store service (step 530). If the update proceeds successfully (~~step~~Yes in decision block 540), the success is reported for the next compilation reply (step 550). However, if either the invocation or the update request is unsuccessful, the attendant software reports failure for the interval B function (step 560). The reports of the interval A and interval B function are included in the compilation reply that is sent to the polling software after the attendant software receives a query from the polling software. ~~FIG. 6~~Figure 7 illustrates the construction of the compilation reply. The interval A and interval B functions ~~60, 62~~ contribute reports ~~64, 66~~ about the last attempt of the attendant~~'s latest attempts~~ software to read mail and update the buffer. The compilation reply further includes, from a service-status function of the service. In addition, a report about the attendant software's login-related capability is included from the login test that is driven via the interval A function. The compilation reply also includes a time value from the server's time indicator. ~~The reply further includes, from a service-status function of the~~ attendant software , a set of reports that includes the status of each of various services running on the server. To include each status in the reply , the attendant software first determines the services for which a status should be reported ~~(FIG.7,~~ as shown in Figure 8 (step 700). The attendant software so determines by referring to a services list that includes services that were pre-selected by the administrator for the particular server. The server's services list preferably includes server messaging software services such as, as noted above, the directory service, the message store service, and the ~~MTA~~MT A service. Depending on the services available on a particular server,

the administrator can configure the server's services list to include other services as well. Examples of these other services are (1) a service that is involved with communicating messages and computer data across the Internet and (2) a facsimile communications service. The attendant software then determines, for each service in the service list, whether the service is functioning properly (step 710). To do so, the attendant software II preferably invokes a facility of the operating system that reports whether the service is functioning properly. The report of the facility serves as the basis of the status determined for the service. Finally, for each service in the services list, the attendant software adds the status of the service to the reply (step 720).

When the polling software receives a compilation reply from a server, the polling software updates the server's clock if necessary ~~(FIG. 8).~~ as shown in Figure 9. To do so, the polling software first retrieves the server's time from the reply (step 800). The time includes a time zone indicator. The server's time is adjusted with respect to time zone and then compared to a master time kept by a master time indicator at the polling software's user computer (step 810). The difference in the times is then compared to a maximum permissible difference that is pre-selected by the administrator (step 820). If the maximum permissible difference is exceeded, the polling software then causes the server's time indicator to be updated to match the user's computer's master time indicator (step 830). By updating each server's clock as just described, the polling software is able to synchronize the clocks of all of the messaging system's servers that share the polling software.

~~Other embodiments are within the scope of the following claims. For example, the attendant may be used to detect not only deficiencies on the server but also usage of software on the server. Furthermore, the attendant may simulate not only a login procedure and a mail-reading procedure, but also other uses of the server, including shared-file access. In addition, various combinations of polling intervals may be used in the escalation procedure, including~~

~~three or more polling intervals or polling intervals using progressively-increasing frequencies or polling intervals that are determined by reference to a prioritized list of deficiencies.~~

The present invention may be embodied in other specific forms without departing from its spirit or essential characteristics. The described embodiments are to be considered in all respects only as illustrative and not restrictive. The scope of the invention is, therefore, indicated by the appended claims rather than by the foregoing description. All changes which come within the meaning and range of equivalency of the claims are to be embraced within their scope.